p. 5

A Hybrid M-algorithm/Sequential Decoder for Convolutional and Trellis Codes *

Fu-Quan Wang
Daniel J. Costello, Jr.
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

June 29, 1990

SUMMARY

The Viterbi Algorithm (VA) is optimum in the sense of being maximum likelihood for decoding codes with a trellis structure. However, since the VA is in fact an exhaustive search of the code trellis, the complexity of the VA grows exponentially with the constraint length ν. This limits its application to codes with small values of ν and relatively modest coding gains. The M-Algorithm (MA) is a limited search scheme which carries forward M paths in the trellis, all of the same length. All successors of the M paths are extended at the next trellis depth, and all but the best M of these are dropped. Since a limited-search convolutional decoder will flounder indefinitely if one of the paths in storage is not the correct one, the data are usually transmitted in blocks. It has been shown that the performance of the MA approaches the

*This work was supported by NSF grant NCR 89-03429 and NASA grant NAG 5-557.

(NASA-CR-186863) A HYBRID M-ALGORITHM/SEQUENTIAL DECODER FOR CONVOLUTIONAL AND TRELLIS CODES (Notre Dame Univ.) 5 p
N90-25528
CSCL 09B
Unclas
G3/61 0293538

VA at high signal to noise ratios (SNR's) with an M which is far less than the 2^{ν} states in the full trellis[1]. Thus the MA can be used with larger values of ν, making larger coding gains possible at high SNR's. However, it still requires a relatively large fixed computational effort to achieve good performance.

Sequential Decoding (SD), on the other hand, can perform almost as well as the VA or the MA with a much smaller average number of computations per decoded branch C_{av}. Simulations show that C_{av} is usually less than 5 when the code rate R is less than the computational cutoff rate R_0 of the channel. However, the number of computations required to decode a block is variable, which may cause an overflow of the input data buffer under severe noise conditions. So, although C_{av} may be quite small, the number of computations required to decode some blocks may be large enough to cause a buffer overflow, which results in an erased block. However, if erasures can be avoided, large values of ν can be used and substantial coding gains can be achieved with SD.

Asenstorfer and Miller[2] proposed a hybrid Viterbi/Sequential decoder which called on the VA to decode particularly noisy blocks which may cause buffer overflow. This method still requires relatively small values of ν, however, since the VA must be used on some blocks. In this paper, we present a scheme which combines SD and the MA. This allows us to use larger values of ν and thus to achieve more coding gain. The proposed algorithm is based on the conventional Fano algorithm (FA) and switches to the MA or a bias-adjusted FA when neccessary. It is shown that when the bias in the Fano metric is small enough, the FA only needs one computation to decode a branch and still provides better performance than a random guess. This property allows us to construct an algorithm which can guarantee erasurefree decoding while maintaining good performance.

To describe the operation of the algorithm, assume the buffer can hold B branches and is divided into three sections which have sizes of B_1, B_2, and B_3, respectively. Let L be the block length, ν be the code constraint length,

μ be the decoder speed factor, j be the number of nonempty buffer sections, and γ be a parameter related to M. We define r_d as the ratio of the number of branches which have already been examined to the length of the block, i.e.,

$$r_d = \frac{n_d}{L+\nu} \tag{1}$$

where n_d is the length in branches from the initial node to the deepest node examined. The algorithm operates as follows:

1. Let $C_{lim} = (L+\nu)(\mu-1)$. Begin decoding with the conventional FA.

2. As long as the number of computations $C \leq C_{lim}$, continue using the conventional FA. Otherwise go to step 3, 4, or 5 according to whether j=1, 2, or 3, respectively. If a terminal node is reached before C exceeds C_{lim}, go to 6.

3. Continue using the conventional FA. If a terminal node is reached before j increases, go to 6. If j increases to 2, check if $r_d \geq \gamma$. If not, use the MA until the block is decoded, and then go to 6. If yes, go to 4.

4. Continue using the conventional FA. If a terminal node is reached before j increases, go to 6. If j increases to 3, go to 5.

5. Use the bias-adjusted FA until the block is decoded.

6. Go to 1 to start decoding the next block or wait for the next block to be received.

The hardware complexity of the above scheme requires both an FA decoder and an MA decoder. Compared to the MA, the complexity of the FA is negligible. So, the hardware complexity is essentially equal to the MA. Since a small speed factor μ can still guarantee very good performance, this scheme is capable of operating at a much higher transmission rate than an MA decoder alone. If the scheme is implemented with a single processor, the advantage is obvious.

Simulations of this scheme and the MA alone with $M = 32$ have been performed on a Sun 3/50 computer. Figure 1 shows the performance of the scheme (referred to as the HMSA) with a rate 1/2 convolutional code of

constraint length $\nu = 12$, $L = 192$ bits, $\mu = 4$, $M = 32$, $\gamma = 0.9$, and of the MA with the same code and $M = 32$. The performance of the VA with a $\nu = 7$ code is also shown in Figure 1 for comparison. It is seen that this scheme can perform better than either the VA or the MA with much less computational effort.

References

[1] C. F. Lin and J. B. Anderson, "M-algorithm Decoding of Channel Convolutional Codes," *Conf. Record*, Princeton Conf. on Information Science and Systems, Princeton, NJ, pp. 362–365, March, 1986.

[2] J. Asenstorfer and M. J. Miller, "A Hybrid Sequential-Viterbi Decoder", Abstracts of Papers, 1986 IEEE International Symposium on Information Theory, Ann Arbor, Michigan, p. 140, October, 1986.

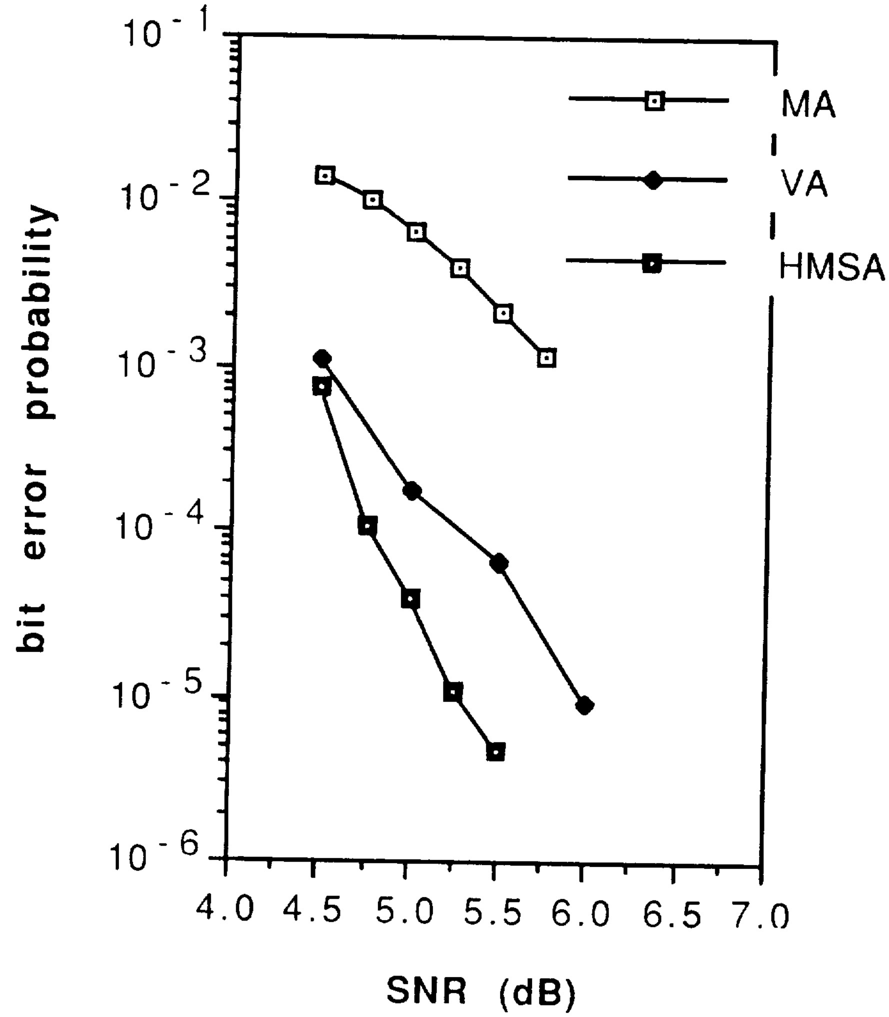

Fig. 1. BER as a function of SNR